



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

BELNET et al

Atty. Ref.: 550-484; Confirmation No. 8029

Appl. No. 10/714,521

TC/A.U. 2136

Filed: November 17, 2003

Examiner: Shiferaw, Eleni A.

For: APPARATUS AND METHOD FOR MANAGING ACCESS TO A MEMORY

* * * * *

March 6, 2008

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

ARGUMENTS IN SUPPORT OF PRE-APPEAL BRIEF REQUEST FOR REVIEW

The claims deal with a data processor that may operate in a secure mode in a secure domain and in a non-secure mode in a non-secure domain (note the four different claim elements). These two domains are kept physically separate so that secure data “is not accessible when [the] processor is operating in a non-secure mode,” as recited in claim 1. A memory management unit (MMU) handles “a memory access request issued by the processor” and performs “predetermined access control functions.” To perform these access control functions, the MMU references descriptors maintained within a table in an internal store that contain access control information for particular memory regions. The internal store also includes a flag associated with each descriptor to identify whether that descriptor is from a non-secure table or a secure table. When the processor is operating in a non-secure mode, the non-secure table is referenced when processing access requests, and similarly when the processor is operating in a secure mode, the secure table is referenced. But in either instance, the MMU initially refers to the internal store to see if the relevant descriptor is stored there. Although not recited in claim 1,

for the panel's information, the flag is used so that the internal store does not need to be flushed every time the processor switches between a secure mode and a non-secure mode or vice versa. See Figure 43 and page 79, line 25-page 80, line 2.

Letwin discloses an operating system for executing programs in a multi-mode microprocessor. The problem addressed by Letwin is one of incompatibility arising due to differences between the Intel 8086 and 80286 microprocessors architectures (see the background text and column 4, lines 37 to 44). In the "real" mode, the 80286 microprocessor architecture emulates the 8086 microprocessor architecture, and in the "protected" mode, the 80286 runs natively. The two processor modes exist in order to provide backward compatibility with an earlier generation of microprocessor architecture. Given this focus on resolving incompatibility problems, it is not surprising that Letwin is not focused on the security problems to which the claims in this application are directed.

Clear Error #1: Letwin Lacks the Secure and Non-Secure Domains

The Examiner equates the non-secure mode" of claim 1 with the real mode in Letwin and the "secure mode" of claim 1 with the "protected mode" of Letwin. But the Examiner fails to identify where Letwin describes the "plurality of domains" or the "secure domain" and "non-secure domain" recited in claim 1. In fact, Letwin fails to teach *both* secure and non-secure modes and secure and non-secure domains. For the claimed domains, the Examiner points to Figures 1-3 of Letwin, but it is not clear how they disclose secure and non-secure domains.

Clear Error #2: Letwin Does Not Make Secure Data Inaccessible in Non-Secure Mode

Claim 1 also recites "said processor being configured such that when executing a program in a secure mode said program has access to secure data which is not accessible when said processor is operating in a non-secure mode." For this feature, the Examiner points to col. 3, line 35 to col. 4, line 58 of Letwin. But Letwin fails to meet the secure data access—non-

secure data access demarcation clearly outlined in claim 1. Column 4, lines 25 to 36 makes clear that a memory space accessibility demarcation in the “protected mode” is not maintained when operating in the real mode. In col. 2, lines 23 to 26, Letwin states that the “real mode” is limited to one megabyte of accessible physical memory. Then in col. 4, lines 25 to 36, Letwin allows the processor in the real mode to address memory locations both above one megabyte (col. 4, lines 28 to 29) and below one megabyte (col. 4, lines 31 to 33). Col. 13, lines 6 to 31 also explains that the issues related to “real mode addressing” above one megabyte are related to address formatting problems and not to security issues. Thus, Letwin fails to teach secure data that “is not accessible when said processor is operating in a non-secure mode” because the real mode can access addresses above 1 MB if a “special technique” is used (see col. 13, lines 9-10).

Clear Error #3: Letwin Lacks the Secure Memory and Non-Secure Memory.

For the same reasons as set forth for Clear Error #2, Letwin also lacks: “a memory configured to store data required by the processor and comprising secure memory for storing secure data and non-secure memory for storing non-secure data.” The “protected mode” memory in Letwin is still accessible in the real mode, which means the “protected mode” memory is not secure memory. Secure data stored in the “protected mode” memory can be accessed in the real mode, which the Examiner equates with the claimed non-secure mode. The Examiner fails to identify where Letwin supports the Examiner’s assertion that “secure data cannot be accessed when mode is unprotected” (sic).

Clear Error #4: Letwin Lacks The Non-Secure Table And A Secure Table

The memory in claim 1 contains “a non-secure table and a secure table, the non-secure table being within the non-secure memory and ... the secure table being within the secure memory.” The Examiner points to col. 4, lines 5 to 16 and Figure 3. Here, Letwin discloses “protected mode descriptor tables to produce a resulting base address identical to that obtained in

real mode” (column 4, line 10 to 12). But Letwin does not disclose a “non-secure table.” The Examiner wrongly asserts that Letwin discloses “unprotected [real] mode descriptor tables.” Instead, Letwin only discloses “memory resident descriptor tables” (col. 10, lines 41 to 42) for use in the protected mode to perform a mapping (col. 10, lines 34-53). In Letwin, there is no need for such a descriptor table in the real mode because in the real mode “all memory addressing is performed in terms of physical or ‘real’ addresses” (col.10, lines 6-9).

Clear Error #5: Letwin Lacks Claim Features Recited for the Non-Secure Table

Because Letwin lacks the non-secure table, Letwin also fails to disclose a “non-secure table being within the non-secure memory and arranged to contain for each of a number of first memory regions an associated descriptor.” For this feature, the Examiner relies on column 9, line 51 to column 10, line 53. Here, Letwin’s description of “real mode” addressing (column 10, lines 6-33) does not refer to tables or descriptors. Letwin discloses “descriptor tables” only in the context of “protected mode” addressing (column 10, line 34 to column 11, line 12).

Clear Error #6: Letwin Lacks a Flag That Indicates a Secure or Non-Secure Table

Claim 1 further recites “the internal storage unit comprising a flag associated with each descriptor stored within the internal storage unit to identify whether that descriptor is from said non-secure table or said secure table.” Because Letwin lacks the claimed non-secure table, it is not possible for a flag to indicate that a descriptor is from a non-secure table. Nevertheless, the Examiner points to column 7, lines 6-27 as supposedly teaching the “flag associated with each descriptor stored” feature. Applicants disagree. This portion of Letwin simply discloses that *a program* written for the microprocessor *includes a flag* indicating whether the program is designed to run in real mode or protected mode, i.e., in the native 80286 architecture or in the emulated 8086 architecture. A flag which determines where the program is stored in main memory cannot be reasonably equated with “an internal storage unit” of the “memory


management unit” “comprising a flag associated with each descriptor stored within the internal storage unit to identify whether that descriptor is from said non-secure table or said secure table,” particularly since Letwin lacks a non-secure table.

Clear Error #7: Letwin Lacks the Claimed Access Control

Lacking a non-secure table and descriptors in the real mode, Letwin does not perform “the predetermined access control functions for the memory access request with reference to access control information derived from the descriptors in the internal storage unit retrieved from the non-secure table.” Letwin’s “protected mode” descriptor tables are used for mapping virtual addresses (col. 10, lines 40 to 43). The Examiner’s reference to the global descriptor table (GDT) and local descriptor table (LDT) is misleading since it is clear from col. 10, line 34 to col. 11, line 58 that these GDT and LDT tables are only used in the protected mode—not the real mode.

Any one of these seven clear errors defeats the anticipation rejection based on Letwin. The final rejection should be withdrawn, and the application passed to allowance.

Respectfully submitted,
NIXON & VANDERHYE P.C.

By: 
John R. Lastova
Reg. No. 33,149

JRL:maa
901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000